PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/60, H04L 12/58 — A1

(11) International Publication Number: **WO 00/13121**

(43) International Publication Date: 9 March 2000 (09.03.00)

(21) International Application Number: PCT/US99/19751

(22) International Filing Date: 27 August 1999 (27.08.99)

(30) Priority Data:

60/098,798	1 September 1998 (01.09.98)	US
09/272,937	19 March 1999 (19.03.99)	US
09/315,732	20 May 1999 (20.05.99)	US
09/351,416	9 July 1999 (09.07.99)	US

(71) Applicant: UNIVERSE COMMUNICATIONS, INC. [US/US]; 2180 Dwight Way, Suite C, Berkeley, CA 94704 (US).

(72) Inventors: DONOHO, David, Leigh; 2830 Buena Vista Way, Berkeley, CA 94708 (US). HINDAWI, David, Salim; 179 Forest Lane, Berkeley, CA 94708 (US). LIPPINCOTT, Lisa, Ellen; 2117 Haste Street #310, Berkeley, CA 94704 (US).

(74) Agents: GLENN, Michael, A. et al.; Glenn Patent Group, 125 Lake Road, Portola Valley, CA 94028 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

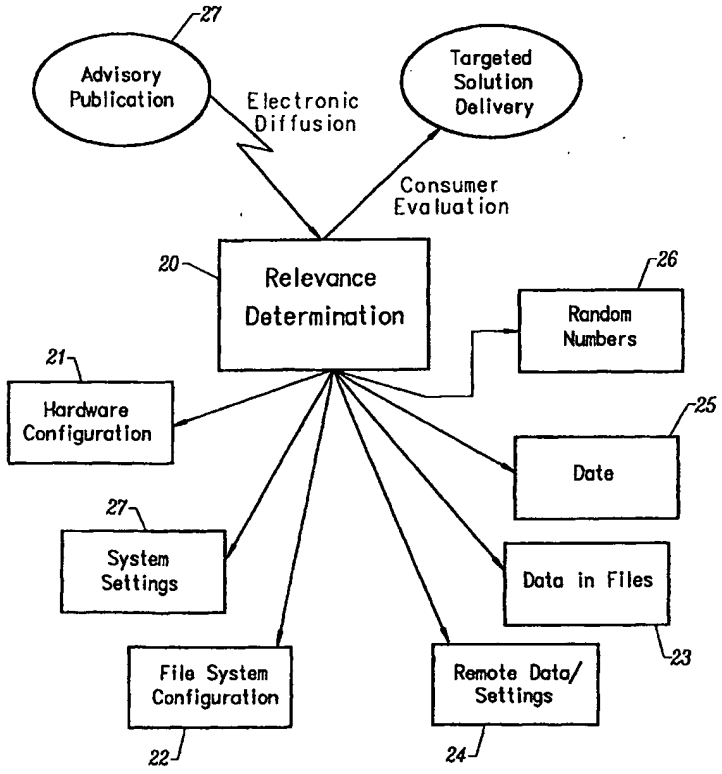
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND APPARATUS FOR COMPUTED RELEVANCE MESSAGING

(57) Abstract

The invention disclosed herein enables a collection of computers and associated communications infrastructure to offer a new communications process which allows information providers to broadcast information to a population of information consumers. The information may be targeted to those consumers who have a precisely formulated need for the information. This targeting may be based on information which is inaccessible to other communications protocols. The targeting also includes a time element. Information can be brought to the attention of the consumer precisely when it has become applicable, which may occur immediately upon receipt of the message, but may also occur long after the message arrives. The communications process may operate without intruding on consumers who do not exhibit the precisely-specified need for the information, and it may operate without compromising the security or privacy of the consumers who participate.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

METHOD AND APPARATUS FOR COMPUTED RELEVANCE MESSAGING

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

The invention relates to a new process of communication using computers and associated communications infrastructure. More particularly, the invention relates to a method and apparatus for computed relevance messaging.

DESCRIPTION OF THE PRIOR ART

The aim of a communications process is to relay information between pairs of actors who, for purposes of the discussion herein, consist of an information provider and an information consumer. The following briefly discusses the concerns of each party.

Concerns of information provider

The information provider knows of pieces of information and of corresponding situations in which certain consumers would find those pieces of information interesting, useful, or valuable. For example, such pieces of information may concern problems consumers who have particular attributes might be interested in solving or that concern opportunities of interest to consumers having such particular attributes. The provider wishes to distribute the information to those consumers in those specific situations.

In principle, an information provider might know of thousands or millions of conditions about which it can offer information. The audience for such conditions might involve thousands or millions of consumers.

A particularly interesting situation is where a typical piece of information should be directed only to consumers having a very special combination of circumstances. A typical piece of information would in principle be of interest to only a small fraction of the consumer base, but where this small fraction nevertheless amounts to large number of consumers.

A challenging but very important case occurs when verifying when the conditions for applicability of a certain piece of information requires knowing a great deal of detailed information about the consumer, his concerns and affiliations, or his property. This information might be considered very sensitive by consumers, who would not want to participate in a process that required disclosure of the information to the provider. Therefore, it might seem impossible to target the information to consumers because only the consumers have access to the information required to make the determination that the information applies to them, and they are unwilling to expend the effort to make a determination themselves, or to give others access to the sensitive information required to make the determination on their behalf.

Concerns of information consumer

The consumer is an individual or organization that knows of information providers who have information of potential benefit to them. The consumer may in fact know of tens or hundreds of such providers. Typically, at any given moment, only a small fraction of the information being offered by the information provider is of potential interest to the consumer. The consumer does not want to review all

the information available from the information provider. He would prefer to see the subset consisting of information, which is relevant to the consumer.

Typically, the information which the provider is offering changes with time and the conditions experienced by the consumer are changing with time. The consumer would prefer not to have to track changes continually in his own status and the status of the information provider's offerings. He would also prefer not to have to remember that pieces of information published some time before could have suddenly become applicable.

The consumer would prefer that a procedure be available for automatically detecting the existence of applicable information as it became applicable, either because the consumer's situation had changed, because the information provider's offerings had changed, or because the conditions for applicability of the information involved time considerations which had become applicable. The consumer would prefer not to reveal to the provider information about his identity or the details of his interests, preferences, and possessions. Rather, the consumer would prefer to receive information in a form where he may carefully study it before using it.

The consumer would also prefer to have a method to inform himself about known problems with an information provider or with a certain piece of information before using the information. Typically, the consumer would prefer that if the decision to use a piece of information is made, the application of the information is painless and essentially automatic. The consumer would prefer to be insulated from the prospect of damage caused by incorrect information.

It would therefore be advantageous to provide a communications technique that addressed each of the above concerns with regard to both the information provider and the information consumer.

SUMMARY OF THE INVENTION

The invention disclosed herein enables a collection of computers and associated communications infrastructure to offer a new communications process. This process allows information providers to broadcast information to a population of information consumers. The information may be targeted to those consumers who have a precisely formulated need for the information. This targeting may be based on information which is inaccessible to other communications protocols, for example because under other protocols the targeting requires each potential recipient to reveal sensitive information, or because under other protocols the targeting requires each potential recipient to reveal information obtainable only after extensive calculations using data available only upon intimate knowledge of the consumer computer, its contents, and local environment.

The targeting also includes a time element. Information can be brought to the attention of the consumer precisely when it has become applicable, which may occur immediately upon receipt of the message, but may also occur long after the message arrives. Again, this is a feature inaccessible under other communication protocols, where the time of distribution of information and the time of consumer notification are closely linked.

The communications process may operate without intruding on consumers who do not exhibit the precisely-specified need for the information, and it may operate without compromising the security or privacy of the consumers who participate. For example, in one implementation, the information provider does not learn the identity or attributes of the individuals who receive this information.

This process enables efficient solutions to a variety of problems in modern life, including the automated technical support of modern computers. In the technical

support application, the disclosed invention allows a provider to reach precisely those specific computers in a large consumer population which exhibit a specific combination of hardware, software, system settings, data, and local environment, and to offer the users of those computers appropriate remedies to correct problems known to affect computers in such situations.

The presently preferred embodiment of the invention is specially tuned to address the concerns of consumers and providers in a technical support application. Many other interesting applications areas and embodiments of the invention are also described herein.

This particular embodiment of the invention is described as follows:

Actors, referred to herein as advice providers, author advisories, which are specially structured digital documents which may contain:

(1) Humanly-interpretable content, such as text and multimedia;

(2) Computer-interpretable content, such as executable programs and data; and

(3) Expressions in a special computer language called the relevance language.

The relevance language describes precise conditions under which a given advisory may be relevant to a consumer, by referring to properties of the environment of the consumer computer interpreting the message, such as system configuration, file system contents, attached peripherals, or remotely accessible data. The humanly-interpretable content in an advisory may describe the condition that triggered the relevance determination and propose an action in response to the condition, which could range from installing software to changing system settings to purchasing information or software. The computer-

interpretable content may include software which performs a certain computation or effects a certain change in the system environment.

Advisories are communicated by a process of publication/subscription over a wide-area network such as the Internet. Advisories are placed by their authors at well-known locations, referred to herein as advice sites. Applications referred to as advice readers running on the computers of advice consumers periodically obtain advisories from advice servers which operate at advice sites.

Advice readers process the messages so obtained and automatically interpret the relevance clauses. They determine whether a given message is relevant in the environment defined by the consumer's computer and associated devices. The user is then notified of those messages which are relevant, and the user may read the relevant advisories and invoke the recommended actions.

Relevance evaluation is conducted by parsing relevance language clauses into constituent method dispatches. These clauses invoke specific inspectors which can return specific properties of the computer, its configuration, its file system, or other component of interest. In effect, the list of properties of the environment which may be referred to in the relevance language and verified by the advice reader is determined by the contents of the inspector library installed at run-time.

The existence of standard inspector libraries provides the advice provider with a rich vocabulary for describing the state of the consumer computer and its environment. In one implementation, the collection of inspector libraries can be dynamically expanded by advice providers.

Advice readers operate continually in an automatic mode, gathering advice from many advice providers distributed across public networks such as the Internet, and diagnosing relevance as it occurs.

Advice readers following an advice gathering protocol, referred to herein as Anonymous Exhaustive Update Protocol, may operate in a manner which fully respects the privacy of the computer's owner. information resulting from the relevance determination, *i.e.* information obtained from the consumer computer, does not leak out to the server. Information on the consumer computer stays on the consumer computer unless the consumer approves its distribution.

Many variations on this specific embodiment are described in detail, including variations which have very different applications, very different message formats, very different gathering protocols, very different security and privacy attributes, very different methods of describing the consumers to whom a message may be relevant, and very different trust relationships between consumer and provider (*e.g.* master-slave relationships). The disclosed invention is shown to be capable of effective embodiment in all these settings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing the process of matching advisories to consumers according to the invention;

Fig. 2 is a block diagram showing an advisor viewpoint according to the invention;

Fig. 3 is a block diagram showing a consumer viewpoint according to the invention;

Fig. 4 is a flow diagram showing a technical support application according to the invention;

Fig. 5 is a block diagram showing an advice site according to the invention;

Fig. 6 is a block diagram showing an advice reader according to the invention;

Fig. 7 is a block diagram showing consumer response to relevance notification according to the invention;

Fig. 8 is a data structure showing an advisory according to the invention;

Fig. 9 is a block diagram showing the process of relevance evaluation according to the invention;

Fig. 10 is a flow diagram showing expression tree generation according to the invention;

Fig. 11 is a block diagram showing named property method dispatch according to the invention;

Fig. 12 is a flow diagram showing an object evaluation model according to the invention;

Fig. 13 is a flow diagram showing an object hierarchy according to the invention;

Fig. 14 is a flow diagram showing a new component of an object hierarchy according to the invention;

Fig. 15 is a data structure showing the contents of an inspector library according to the invention;

Fig. 16 is a block diagram showing situational advice according to the invention;

Fig. 17 is a block diagram showing simulated conditions according to the invention;

Fig. 18 is a block diagram showing a commodity market according to the invention;

Fig. 19 is a flow diagram showing a relevance-adapted document according to the invention;

Fig. 20 is a flow diagram showing questionnaire processing according to the invention;

Fig. 21 is a flow diagram showing a mandatory feedback variant according to the invention;

Fig. 22 is a flow diagram showing a consumer feedback variant according to the invention;

Fig. 23 is a flow diagram showing masked bi-directional communication by an anonymous server according to the invention;

Fig. 24 is a flow diagram showing a further mandatory advice variant according to the invention; and

Fig. 25 is a block diagram showing remove relevance invocation according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention implements a process of communication which systematically solves the problem of linking an information provider to information consumer. The invention provides a system which depends on the use of computational devices connected by communications networks. In actual practice, these devices could range from traditional large-scale computers to personal computers to handheld personal information managers to embedded computational devices in the ambient environment, including consumer appliances such as remote controls and smart TVs, or other common computationally-dense environments, such as transportation vehicles. The communications mechanisms could include a modem or other wired media, or wireless communications, using the Internet or other protocols, and could include the physical distribution of media. Whatever the specific instance, for purposes of the discussion herein, the computational device shall be referred to as a computer and the communications infrastructure shall be referred to as a network. Typical examples of such infrastructure include intranets (private computer networks), and the Internet, the large public computer network that hosts the World Wide Web and related services.

The invention architecture is best understood if a specific terminology is adopted, which evokes a focused instance of the above described communications problem. The specific units of information to be shared henceforth are referred to as pieces of advice (see Fig. 1). The special digital documents conveying advice are referred to as advisories. An advice provider 10 is an organization or individual which offers information in the form of advisories 12a-12d. The provider is represented by a server computer in a communicating network of computers. An advice consumer 14a-14c is an organization or individual which receives information in the form of advisories. The consumer is represented by a

computer referred to as the consumer computer in a communicating network of computers.

It is helpful to think in concrete terms, and to suppose that the advice provider is in fact a large organization running a large-scale server computer; that the advice consumer is in fact an individual represented by a single personal computer, smart TV, personal information manager, or other personal computational device; and to suppose that the network of computers may communicate according to a protocol similar to the TCP/IP protocol now in use by the Internet. In actual practice, many variations can be expected. For example, an advice provider may constitute an individual represented by a personal computer, an advice consumer may be a corporation represented by a large-scale computing engine, and the communications process underlying the invention may be realized with other protocols operating over other physical means of communication.

Using this terminology, it is now possible to describe a key purpose of the invention. The invention allows one to relay advisories from advice providers to advice consumers. The communications protocol allows narrowly-focused targeting by automatically matching advisories with consumers for whom those advisories are relevant.

Relevance determination (see Fig. 2) is carried out by an applications program, referred to as the advice reader 20 which runs on the consumer computer and may automatically evaluate relevance based on a potentially complex combination of conditions, including:

- Hardware attributes. These are, for example, the type of computer on which the evaluation is performed, the type of hardware configuration 21, the capacity and uses of the hardware, the type of peripherals attached, and the attributes of peripherals.

- Configuration attributes. These are, for example, values of settings for variables defined in the system configuration 22, the types of software applications installed, the version numbers and other attributes of the software, and other details of the software installation 27.

- Database attributes. These are, for example, attributes of files 23 and databases on the computer where evaluation is performed, which may include existence, name, size, date of creation and modification, version, and contents.

- Environmental attributes. These are, for example, attributes which can be determined after querying attached peripherals to learn the state of the environment in which the computer is located. Attributes may include results of thermal, acoustic, optical, geographic positioning, and other measuring devices.

- Computed attributes. These are, for example, attributes which can be determined after appropriate computations based on knowledge of hardware, configuration, and database and environmental attributes, by applying specific mathematico-logical formulas, or specific computational algorithms.

- Remote attributes 24. These are, for example, hardware, configuration, database, environmental, and computed attributes that are available by communicating with other computers having an affinity for the consumer or his computer.

- Timeliness 25. These are, for example, attributes based on the current time, or a time which has elapsed since a key event, such as relevance evaluation or advice gathering.

- Personal attributes. These are, for example, attributes about the human user(s) of the computer which can either be inferred by analysis of the hardware, the system configuration, the database attributes, the environmental attributes, the remote attributes, or else can be obtained by soliciting the information directly from the user(s) or their agents.

- Randomization 26. These are, for example, attributes resulting from the application of random and pseudo-random number generators.

- Advice Attributes 27. These are, for example, attributes describing the configuration of the invention and the existence of certain advisories or types of advisories in the pool of advice.

In this way, whatever information is actually on the consumer computer or reachable from the consumer computer may in principle be used to determine relevance. The information accessible in this way can be quite general, ranging from personal data to professional work product to the state of specific hardware devices. As a result, an extremely broad range of assertions can be made the subject of relevance determination.

The advice reader 30 (see Fig. 3) may operate automatically to determine relevance. It may present to the consumer a display of relevant advisories 32 only from several advice sites 33a-33c, so that the consumer is not burdened with the task of reading irrelevant advisories. In this way advisories may provide

an automatic diagnosis 34 to any problem which a relevance clause may describe.

Advisories are digital documents which may contain an explanatory component, describing in terms the consumer can easily understand the reason that the advisory is relevant and the purpose and effects of the action which is being recommended to the consumer. These digital documents may also contain, as another component, executable computer programs, or links to executable computer programs. In this way advisories may provide an automatic solution to any problem which the relevance message may have diagnosed, and which may be activated at the consumer's discretion.

In short, the invention posits a situation where proactive advice providers identify situations of interest to consumers and provide advice about dealing with such situations.

Computer Technical Support Application.

To make the above generalities more concrete, a particular application area is described where this communications process may be of considerable utility (see Fig. 4).

In the technical support application, the advice provider offers a computer-related product or service, such as hardware, software, Internet service, or data processing service. The advice provider has a potentially large, potentially widely distributed customer base 40. In part from user input 42, the advice provider knows of problematic situations 41 which may affect certain computers belonging to the customers. The advice provider identifies these problematic situations 43, which may include the use of out-of-date versions of software, improper system settings, conflicting combinations of software applications,

inadequate physical resources, corrupted files, other similar phenomena. The advice provider may know, for each problematic situation, a precise combination of hardware, system configuration, database configuration, timeliness, and other attributes which may signal the situation. The advice provider may know a precise solution 44 to each problematic situation, which may include:

- A suggestion to the user to modify usage patterns;
- A suggestion to the user to read a document;
- A proposal to upgrade to a new software version;
- A proposal to modify system settings;
- A proposal to run a certain script to effect a solution; or
- A proposal to download and execute special applications to correct the situation.

The advice provider authors an advisory 45, which is then preferably tested 46, and made available to relevant users at an advice site 47. In this way, the advice provider can use invention to reach the consumer population efficiently. The provider packages the information about the specific situation as a formal advisory concerning the situation. This digital document may include:

- A precise formal-language specification of conditions under which the situation occurs;

- Explanatory information intended for consumers who are in the given situation, describing to those consumers the situation they are in, the implications of the situation, and the providers proposed actions to correct the situation; or

- Digital content providing automatic solution or response.

The advice provider publishes the advisory 40 over the Internet or an Intranet, through an advice server running at the provider's advice site. For example (see Fig. 5), the advice site may comprise a directory of advice files 51a-51b and inspector files 52a-52b (discussed below). These advisories may be communicated to the outside world 54 via such media as a directory message server 55, an HTTP server 56, and FTP server 57, or a file server 58.

The advice consumer is a user of the products and services of the advice provider who knows of the advice provider's advice site and generally trusts the provider's organization and the advice that it authors. The advice consumer has available on his computer the advice reader application. The advice consumer instructs his advice reader to subscribe to the advice site offered by the advice provider.

The advice reader 20 (see Fig. 6), at scheduled intervals or under user manual control via a user interface 65, gathers advisories to which the user subscribes. Subscription to advisories are entered with a subscription manager 67 based, at least in part, on information in various user site definition files 68. Advisories are gathered from the advice provider's advice sites 33a-33b using a gatherer 60. The reader then parses the advisories using an unwrapper 61 and adds these advisories to any already existing body of advisories. Advisories may be provided to the reader via any of several sources, including alternate input streams 62. The advice reader determines the relevance of any of the existing or

new advisories with a relevance evaluation module 63. This determination is made either continuously, at scheduled intervals, or under user manual control. The advice reader includes a user interface 65 that receives relevant advisories and a display and management system 66 that displays relevant advisories for inspection by the consumer the relevant advisories. In some embodiments of the invention, an advisory may also be subject to digital verification using a verification module 64 (discussed in greater detail below).

A typical relevant advisory is reported to a consumer as follows:

> Your computer has a certain combination of hardware and software and settings. Computers with this combination have frequently been reporting a particular problem. Our company has a solution. It will change your computer settings. If you accept to use this solution, your problem will go away. This solution has been rigorously tested before release, and represents our best known way of dealing with this problem.

The advice consumer reviews such relevant advisories 100 (see Fig. 7), and acts on the advisories 110, for example by ignoring the advisory 111. Otherwise, the user potentially deliberates, which deliberation may include informing himself further about the advisory or its author 112, informing others of the advisory 113, or taking some other offline action 114 and then, depending on the outcome of the deliberation, he approves or denies approval. If the consumer gives approval, an automatic solution may result, which may involve a variety of activities, including software downloading 72, installation, and execution 71, an automatic electronic response 73, or the purchase or order of a digital object 70.

This particular application area shows how invention can be used to diagnose and fix problems on a computer automatically. There are many other applications

areas of the invention, which may involve making commercial transactions rather than fixing computer problems, or offering new forms of private communications.

Responsiveness to Concerns

The invention is fully responsive to the concerns discussed above.

Provider Concerns

Large Scale Communications. In common with other computer-mediated communications systems, such as the world-wide web, the invention is able to reach a large number of consumers and convey to them a large body of informational messages, at low cost.

Automatic Operation. The matching of information to consumers is done without the need for case-by-case intervention of skilled human operatives.

Exclusive Targeting. The invention enables information to flow precisely to the appropriate consumers. The provider can guarantee this by carefully specifying the conditions under which a piece of advice is relevant.

Targeting with Intimate Knowledge. Information targeting in the invention is precisely focused on the attributes of the consumer because it has access to intimate knowledge of the inner details of the consumer computers state, without necessarily disclosing this knowledge to the provider. This degree of targeting is not possible under other protocols because other protocols require disclosure of this information to the provider to determine if a piece of information is relevant.

Consumer Concerns

The invention satisfies the main consumer concerns mentioned earlier.

Automatic Unattended Operation. The invention is an automated messaging system which operates successfully with infrequent consumer involvement. The advice reader can periodically gather new advice from advice sites that it subscribes to. This process may be fully automatic (manual intervention is also available). The databases of advice resident on the consumer computer may be continually evaluated for relevance by automated unattended operation of the advice reader.

Provision of Narrowly Targeted information. In a typical mode of operation, the consumer only sees information relevant to his precise attributes, including attributes derivable from the contents of his computer, associated peripherals and affiliated computers.

Timely Provision of information. In a typical mode of operation, a piece of advice may enter the consumer computer and remain resident for an extended period of time before becoming relevant. information is displayed when it has become applicable, not before it does.

Opportunity for Deliberation. Typically, the advice reader does not automatically apply a recommended solution operator. Rather, the advice reader gives the consumer the chance to study the diagnosis and recommendation, and to evaluate the credibility of the provider, before proceeding. There are three special aspects to the deliberation process available in invention:

- Disclosure of Potential Risks. By exploiting known user interface methods, such as HTML display with hypertext links, the invention enables advice

providers to inform consumers fully about potential risks associated with following a certain recommended course of action.

- Discovery of Consumer Complaints. Via devices to be discussed below (such as the Better Advice Bureau) consumers may use the advisory mechanism to inform themselves about the existence of known and foreseeable privacy and security risks associated with specific advisories and/or advice providers before accepting proposed solutions.

- Correction of Known Defects. The invention allows advice providers to retract their own faulty advice. An instance of this is the UrgentAdviceNet mechanism (discussed below) for rapidly distributing advisories to the invention population.

Automated Solution. Typically the advice provider authors an advisory in such a way that the advice reader offers it to the user to apply a recommended solution operator automatically after the user has given approval. Thus, the invention offers an automated solution to the user's condition under user guidance.

In short, the invention provides a mechanism to match consumers with highly specific relevant advisories efficiently in a communications structure which is responsive to consumer concerns.

Security and Privacy Technique: One-Way Membrane

The disclosed invention offers a comprehensive process for computed-relevance messaging. This is a broad idea, with many possible applications. In certain settings, this type of messaging must be implemented in a fashion which pays special attention to security and privacy concerns, *i.e.* a one-way membrane 35

(see Fig. 3). For a concrete instance, consider the technical support application (discussed above), where:

- Communication must take place over public networks such as the Internet;
- The advice provider is a large business or other concern; and
- Advice consumers make up a widely distributed group of lay users.

In this setting, consumers have special concerns about any process which functions as if it had intimate knowledge of the consumer's computer and its contents. These concerns are legitimate because the Internet is widely known as an insecure communications medium. Hence, systems which interact with the Internet, and which appear to function as if they had intimate knowledge about a user, might appear to enable privacy intrusions.

The invention addresses this problem by proposing a method of interaction between the consumer computer and the Internet which protects the consumer's privacy. This mechanism need not be used in other settings. For example, in certain private computer networks, commonly referred to as intranets, the invention has a variety of applications. In such settings, security and privacy are considered guaranteed by physical control of the computer and communications infrastructure involved, and possibly by contracts creating obligations on the participants in the process.

The invention employs a special protocol for subscription and gathering in the security and privacy critical setting. For purposes of the discussion herein, this setting is referred to as the Anonymous Exhaustive Update Protocol (AEUP). The intention of this interaction protocol is to create a one-way membrane, where information can enter the consumer computer in the form of advisories, but

information about the consumer does not leave the consumer computer unless it is the consumer who initiates the transfer.

The AEUP protocol is described as the default protocol of the invention. The reasons that this protocol offers consumers privacy is discussed below. This document also describes many applications where security and privacy are not critical to acceptance by the consumer. Thus, it is possible to provide a certain degree of security and privacy protection without using this protocol. See below for a discussion of alternative protocols, such as the Anonymous Selective Update Protocol (ASUP).

A comprehensive discussion of privacy and security concerns is given below. The invention addresses:

- Consumer Privacy Concerns. The invention fully respects consumer privacy concerns. In an implementation offering AEUP, consumers may benefit from narrowly-targeted advice without ever needing to reveal their identity, nor any of the attributes that were checked in determining relevance, nor the fact of relevance itself.

- Consumer Initiative. In a typical mode of operation, no advice is received by the advice reader unless the consumer initiated the subscription. This protects the consumer from unwanted communications.

- Privacy of Automatic Operations. Under AEUP, the operation of gathering advice from sites, the operation of evaluating relevance, and the operation of displaying relevant advice to the consumer need not result in the disclosure of consumer data to the advice provider.

- Frustration of Intrusions. Certain embodiments of the invention contain mechanisms, described below, to prevent compromises of privacy even in case of certain illegal eavesdropping activities

- Consumer Security Concerns. The invention fully respects consumer security concerns. In an implementation offering AEUP, consumers may benefit from narrowly-targeted advice without exposing themselves to security threats from malicious sources.

- Consumer Initiates Subscriptions. In a typical mode of operation, no advice is received by the advice reader unless the consumer initiated the subscription. The process of subscription to an advice site connotes limited trust by the consumer for the provider. Hence, in typical operation, advice is only received from trusted sites.

- Harmlessness of Automatic Operations. Typically, the process of gathering and evaluating advisories has no noticeable effects on the computer system. Any recommended solution is applied only upon prior notification of the user and subsequent approval. Consumers who use invention to merely peruse relevant messages, but do not follow the recommended actions, face no significant risk.

- Disclosure of Potential Risks. By exploiting known user interface methods, such as HTML display with hypertext links, the invention enable advice providers to inform consumers fully about potential risks associated with following a certain recommended course of action.

- Discovery of Consumer Complaints. Via devices that are discussed below (such as the Better Advice Bureau), consumers may use the advisory

mechanism disclosed herein to inform themselves about the existence of known and foreseeable privacy and security risks associated with specific advisories and/or advice providers before accepting proposed solutions.

- Correction of Known Defects. The invention allows advice providers to retract their own faulty advice. It allows other people to criticize an advice providers faulty advice.

- Automated Solution. The advice provides typically authors an advisory in such a way that the advice reader offers to apply a recommended solution operator automatically to the user system after the user has given approval.

Thus, the invention provides a mechanism for efficiently matching consumer with highly specific relevant advisories in a communications structure which is responsive to consumer concerns.

Layers of Invention

The present document describes computed relevance messaging from many viewpoints, *i.e.* from one extreme of a general communications process to the other extreme of a set of specific protocols that have been implemented by Universe Communications, Inc. of Berkeley, California. It is worthwhile to classify the several layers of the invention as described herein:

Relevance Guided Messaging. The general communications process used by the invention has five elements (see Fig. 8):

- A Relevance Clause 80. An assertion about the state of a consumer computer, its contents, or environment which can be automatically evaluated by comparing the assertion with the consumer computer's actual state.

Typically, the relevance clause is preceded by a subject line 82 which gives a general description of the advisory's subject matter.

- An Associated message 81. A message or messages associated with the clause whose suitability for the consumer is determined at least partially by the evaluation of the clause.

- A Gatherer 60 (see Fig. 6). An application that sees to it that relevance clauses flow into the consumer computer from various locations, perhaps by regular synchronization.

- A Watcher 63 (see Fig. 6). An application that has the ability to evaluate relevance clauses, *i.e.* assertions about consumer computer's own environment, by comparing them with the actual state of the environment, and by inspecting properties of the consumer computer and its environment and checking if these point towards or away from relevance.

- A Notifier 65, 66 (see Fig. 6). An application that has the ability to display messages to a user under at least partial guidance of an evaluated relevance clause.

A key difference of the invention from other targeted information providers is that the invention provides a detailed tool for tapping into very highly defined targets, which other protocols for targeting information cannot match because they do not routinely have access to the state of the consumer's environment.

The details of relevance guarded messaging are less important than this five-part model. For example, in one implementation, the five-part model is run on a computer network in a secure network such as a corporate intranet. In another

implementation, the five-part model is run on a public computer network such as the Internet. Certain concerns that affect the public setting (*e.g.* security and privacy) might be completely irrelevant in the private setting, where those concerns are addressed by the physical control of the network. In either setting, the basic five-part model of relevance guarded messaging makes a valuable contribution to connecting providers with consumers.

It is important to note that this five-part model may have embodiments in which these five parts are not immediately evident. Potential implementations which make it clear that there can be many superficially different ways of achieving this basic structure are described below. For example, the relevance clause and the associated message may be packaged together in the same file and communicated simultaneously. In a different embodiment, the relevance guarded message can be communicated in two stages, where the first stage sends a relevance clause, and the second part is sent only if the first part leads to a relevant result and if the consumer computer asks the provider for the second part. Conceptually, the same useful effect can be obtained using either of these two messaging protocols. Both methods are embodiments of the same invention.

Relevance Guarding with Security and Privacy. Owing to the tremendous importance of public networks, such as the Internet, an implementation of the five-part model which also addresses fundamental privacy and security concerns is of great significance. The mechanism by which the basic five-part model is extended (*e.g.* through AEUP, ASUP, or substantially equivalent protocols) to become a secure and private system over public networks is an important embodiment of the disclosed invention. It is potentially helpful for the broad consumer acceptance of computed relevance messaging.

Preferred Embodiment of the Invention. The presently preferred embodiment of the invention consists of a large collection of different interacting components,

carefully designed to meet the goals underlying this system. The many subsystems illustrate the potential of the invention in the technical support application. Those skilled in the art will appreciate that there are many other applications to which the invention may be put.

Variant Implementations. The specific implementation was arrived at after a long series of different application areas were examined and carefully studied. This document describes in considerable detail a large number of variant implementations modify the basic operation of the central implementation for other market areas or other demands. For example, in certain settings, the use of low communications bandwidth is important and privacy is unimportant. A variation for that setting is discussed below.

Invention Components

The following discussion describes the key components in what is currently regarded as the best mode of implementing the disclosed invention. In this implementation it is assumed that communications are via standard Internet techniques, and that the advice provider and advice consumer are both relying upon standard network connected computers.

Advice Provider Components

The following is a listing of component names, followed in various subsections by a brief discussion of each component:

- advice site
- advisories

- site signature
- site description file
- inspector library files
- supplementary files

While these general components may be implemented in many ways, it easiest to describe their form and function in the currently understood best mode, based on the use of Internet communications protocols. Those skilled in the art will appreciate that this is not the only possible implementation.

advice site

This is a standard place on the Internet (see Fig. 5), *e.g.* a URL-addressable directory on a server computer, combined with server software that responds to certain TCP/IP requests for information.

The site directory may contain a plurality of files, including advisories, digests of advisories, and inspector libraries.

The software associated with the server may perform the functions of an HTTP server, an FTP server, or a file server, thereby providing access to the files stored in the directory using well-known communications protocols. The software associated with the server may also perform the functions of a specialized server, implementing invention-specific communications protocols.

These protocols may include:

- The ability to serve a directory message describing the contents of the site directory, including filenames, sizes, and dates;

- The ability to serve an abstract message which describes in abbreviated form the contents of the files in the directory;

- The ability to engage in security handshaking;

- The ability to perform challenges to advice readers to validate their authenticity; and

- The ability to meter traffic through the site, and compute summaries of traffic levels.

The function of advice site server software is to process certain requests made by an advice reader running on a consumer computer. The advice reader may request information about the directory of the site, may ask for abstracts of advisories, and may ask for contents of individual advisories. The transaction between advice server and advice reader is described further below.

Advisories

The advisories in an advice site are digital files. Advisories typically have some of the following components:

- A relevance precondition written in a formal relevance language, which is used to describe attributes of a computer and/or its contents and/or its environment. For more information on the relevance language, see below.

- A humanly-intelligible component which may summarize the purpose of the message, may describe the author, may explain the precondition in human language, and may explain the solution in human language.

- A computer-intelligible component which potentially offers either software tools to solve the problem or Internet access to software tools solving the problem. In the currently understood best method for this implementation, an advisory is a specially formatted ASCII file built using the MIME Internet standards track specification documented in RFC 1521 *et seq.* (see N. Borenstein, N. Freed, *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*, Internet Standards Track RFC 1521(1993)). This format is currently used for transport of Internet mail; it contains headers documenting the sender of the message and its subject, and mechanisms for including digital signatures. A MIME file is easily transported over the Internet and is easily broken into its constituent components using parsing algorithms well-known in the Internet community. The advisory file format is described further below (see, also *A Guide to Writing Advisories for AdviceNet*, Universe Communications, Inc., Berkeley, CA. (1998)).

Authoring Advisories.

Site Signature

Associated with an advice site may be a certain digital signature mechanism, for example one of the standard signature mechanisms using public-key/private-key pairs. The signature mechanism may be used to sign advisories in a fashion that allows advice readers to verify that the advisory was in fact authored by the advice provider.

Site description files

The site description file (SDF) is a specially structured ASCII text file authored by the advice provider. It describes the provider's advice site and serves as the basis for a consumer to initiate a subscription. This file specifies the site location (URL), the site name, and site security characteristics, such as whether the site avows only advice which has been digitally signed. It also provides various parameters of the subscription process intended for use by the advice reader (for example, the recommended frequency of synchronization, and the type of subscription relationship (free/fee)). It may contain humanly interpretable text indicating the purpose of the site.

The SDF may also contain the public key associated with advice authored by the site. This public key is needed to verify signatures on advice authored by the site.

The SDF may also be signed by a trusted authority, to establish the authenticity of this site description file. For example, it may be signed by advisories.com or the Better Advice Bureau: see below.

The SDF may also contain a ratings block, provided by a trusted ratings service, to establish trust in the respect for privacy and security and the usefulness of advice at this site. See, for example, below.

inspector libraries

inspector libraries are libraries of special purpose executable code, which may be accessed by advice readers for the purpose of extending the capabilities of the relevance language. In effect, inspector libraries provide a mechanism for advice site specific extensions to the relevance language.

Supplementary files

The contents of the advice site discussed so far play important roles in the ordinary conduct of the invention. In one typical implementation, additional files may be present in the advice site directory. In such an implementation, data and applications files which do not play a role in the conduct of the invention *per se* may be included in the advice site directory. These files are distributed as are other files at the advice site. This implementation allows the distribution of installers, uninstallers, shell scripts, JAVA, and Visual Basic programs, *i.e.* in general, packages of data, applications, and other resources, that may play a supporting role in evaluating and following advice issued at the site. For example, such additional files may play a role as databases searched by the advice provider's own inspector libraries or as applications used in implementing the advice providers recommended solutions.

Advice Consumer Components

The following is a listing of component names from the advice consumer perspective, followed in various subsections by a brief discussion of each component:

- advice reader

- subscription database
- advice database
- user profile
- inspectors
- solution wizards
- advice reader

The advice reader is an application running on the consumer computer. It is responsible for liaison with the advice site and for managing interactions with the user. The advice reader maintains a directory of files on the consumer computer. Inside that directory are contained various files described below which are used/managed in the course of advice reader operation.

The advice reader has a number of jobs, which are listed below without elaboration:

- Manage subscriptions
- Synchronize with advice site
- Gather advisory files
- Unwrap advisory messages

- Manage advice Database
- Manage relevance Evaluation
- Evaluate relevance of Individual advisories
- Invoke inspectors
- Display relevant advisories to User

The process is described in detail below.

Subscription Database

The advice reader maintains a database of subscription information which allows for the scheduling and conduct of site synchronization by the gatherer component. The subscription database contains information about the address of the advice site; information and recommendations provided by the advice sites site description file, such as recommended frequency of synchronization; information needed to verify digital signatures associated with the advice site; and information associated with the users experience with the advice site.

Advice Database

The advice reader maintains a database of advice that has been received from various advice sites. These may be indexed according to the site from which they were received according to the systems that the advice concerns, or according to other principles which would be helpful to the consumer or to the author.

The advice reader may organize advice into pools of advice which share a common basis for treatment. Examples of this principle include a pool of advice specially targeted to the concerns of one user of a multi-user consumer computer, a pool of advice scheduled for manual relevance evaluation only, and a pool of advice scheduled for nightly evaluation at a certain time.

User Profile

The advice reader maintains a special file or files containing data which have been obtained from interviews with the user, deduced from his actions, or deduced from the properties of the computer or its environment. Such data may describe the computer or its environment, and may also describe preferences, interests, requirements, capabilities, and possessions and plans of the user, including things unrelated to computer operations.

The file or files may be encrypted. The file or files may be organized by advice site so that they describe interests, preferences, and so forth to be accessed by relevance queries associated with a specific site only.

inspectors

inspector libraries contain executable code which may be invoked by the advice reader as part of the relevance evaluation process. Inspectors can examine properties of the consumer computer, storage devices, peripherals, environment, or remote affiliated computers. These are further described below.

Solution Wizards

Solution wizards support the process of automated solution. They are applications which can perform stereotyped functions that are frequently of use for solving problems on computers. These are described further below.

Transaction Overview

The following discussion describes the basic model for an Internet-based transaction using the invention.

Subscription Model

In the invention, the initiative to begin an interaction typically comes from the consumer. The consumer becomes aware of the existence of an advice provider and associated advice site(s), for example, as part of installing a new hardware or software product on his computer, or as a result of advertising, or sharing experiences with other consumers. The consumer, after potentially informing himself about the kind of advice being offered at that site and its reliability, makes a decision to subscribe. The consumer, interacting with a piece of the advice reader called the subscription manager 67 (see Fig. 6), configures the advice reader to subscribe to the given advice site, by supplying it with either the corresponding site description file 68, or with a pointer to such a file, or with a pointer to the site itself which contains an instance of such a file. The consumer, after studying the terms of interaction recommended in the SDF, configures the parameters associated with the subscription, which control how frequently advice from the site is gathered.

Advice Gathering Using AEUP

Periodically, under the terms of the subscription, or manually under user control, the advice reader initiates a site synchronization. A component of the advice reader, referred to as the gatherer, has the duty to synchronize the consumer site image with the current image of the advice site. These states can be different if the advice site has retracted advice or authored new advice since the most recent synchronization. The gatherer makes sure that there is a one-to-one correspondence between advisories at the advice site and advisories in the consumer machine. The gatherer opens a connection to the directory message server at the advice site. After an optional security handshake to verify the authenticity of the advice reader and server, the gatherer queries the server for a directory message. The gatherer inspects the response and checks whether the site directory has changed since the previous synchronization. If not, there is no need to obtain any files from the advice site, and the session may end. If the directory has changed, or if this is the first synchronization ever, the gatherer initiates FTP and/or HTTP and/or file server access to the new files. The gatherer also deletes any advisories on the consumer computer which no longer correspond to advisories on the server, and this terminates the synchronization of the consumer site image with the true site image.

The protocol just described is the AEUP protocol that is described above. The gatherer is allowed, by the advice server, to gather all the files at the advice site anonymously or, at any rate, all files which have not previously been gathered. The intention is that the advice stored on the consumer machine consists at any given moment of all the advisories offered at the advice site at the time of the last synchronization, other than those that the user has specifically deleted. Hence, there is no selective gathering. Rather, gathering is exhaustive, *i.e.* every piece of advice is gathered. The implications of this protocol and alternative protocols are discussed below.

Unpacking Advisories

As described below, an advisory file is a potentially complex hierarchical structure, which may contain one or more than one message. The advice reader unpacks all the components of this structure. Components of the structure may be signed using a digital signature method, *i.e.* at unpacking time those signatures are verified. After unpacking, the advisories are entered in a pool of all advice, old and new, to be evaluated. In one typical implementation, the invention may suppress entry into the system of unsigned advisories or of advisories whose signatures cannot be verified.

Relevance Evaluation

As a matter separate from gathering, the pool of all advice to be evaluated may be processed, either continuously, or according to a consumer-defined schedule, or an immediate user request, or some specified trigger event (see Fig. 9). The advice reader parses the individual message and identifies the clauses determining relevance. These clauses are expressions in the formal relevance language which is described below. The advice reader parses the clauses using an expression tree generator 91 into a tree of elementary subexpressions (see Fig. 10) and then evaluates each subexpression of the tree using an expression tree evaluator. If evaluation proceeds successfully and results in a value of True, the message is deemed relevant 93. A dispatch method 94 is then used to consume the advisory which may include a file system inspector that identifies appropriate directory and file name references 96 in various user volumes 97, 98; a registry inspector 99 that inspects an operating system registry 120; an operating system inspector 121 that inspects various system elements 122; or a hardware device inspector 123 that inspects various system devices 124.

Inspectors

Evaluation of subexpressions is performed by methods called inspectors (see Fig. 11) which may perform mathematico-logical calculations, execute computational algorithms, return the results of system calls, access the contents of storage devices, and query devices or remote computers. These methods are called inspectors because a frequent purpose is to inspect the properties of the consumer computer, its configuration, or contents of its storage devices. Inspectors may come built in to the reader, and may also be plugged in via DLL or similar mechanisms. Thus, an object 130, property name 131, and/or string selector 132 is dispatched to a reader using a method dispatch module 134 in accordance with dispatch information contained within a method dispatch table 133. Various inspectors 135, 136 are provided at a user location, each of which includes an inspector library 137, 139 and associated methods 138, 140. Inspectors are described in greater detail below.

User Interface

After relevance has been decided for an item in the advice pool, a relevant item may be entered into a list of items to be displayed. This list may be displayed to the consumer according to typical user-interface models. The user-interface may inform the user about the author of the advisory, about the date the advisory was acquired, about the date the advisory became relevant, about the subject of the advisory, and about other attributes of the advisory message. The user interface may offer the user to display the explanatory content of individual advisories. Depending on the advisory, the explanatory content may contain simple text explanations, or may contain more elaborate multimedia explanations. Depending on the advisory, the explanation may identify the situation which caused the advisory to be relevant, the implications of relevance, the recommended action or actions to take at this point, the anticipated effects of

taking those actions or of not taking them, or the experiences of other users or other organizations with the proposed actions. The user studies this explanatory content, perhaps performing additional research (for example studying the trustworthiness of the provider, or the opinions of other users).

Recommended Response

As part of the display of a relevant advisory, the user is typically offered the possibility of an action in response to the situation. Possible outcomes include:

- consumer ignores information/proposal. The consumer reviews the advisory, decides he does not wish to pursue it, ignores the content, and deletes the advisory.

- consumer is notified. The consumer reviews the advisory, or some other document it refers to, and learns something important or interesting.

- consumer is entertained. The consumer reviews the advisory, or some other document it refers to, or some multimedia content it contains, or some multimedia content it refers to, and is exposed to a stimulating presentation.

- consumer forwards information to another. This may include friends, family, colleagues, or associates. Forwarding may involve off line transport or electronic transport, such as e-mail.

- consumer initiates correspondence with provider or other. This may include contact by mail, phone, fax, or e-mail. This may also include participation in an information exchange, including for technical support, training, or market

survey purposes, as well as participation in a sale or other commercial interaction.

- consumer initiates on-line participation in a timely event.

- consumer purchases object by e-commerce. This may include a purchase entered by clicking on a button in the advice reader window which entry to e-commerce mode.

- consumer fills out a form. This may include a form rendered by a Web browser, or a text file form intended to be returned by e-mail, or a form intended to be filled out and faxed or mailed back.

- consumer initiates off line action in real world. This may include any off line action ranging from actions associated with the computer modifying the state of hardware devices, gathering information in the environment surrounding the computer, or reading some instructions in a manual before beginning an online process. This action may also include purely personal items.

- consumer modifies system setting or data field on computer. This may involve the consumer executing a series of manual operations on the computer to change settings of some system component or software application or to modify an entry in a database.

- consumer initiates an Install/Uninstall/Execute solution. This may involve the consumer clicking on a button in the advice reader, followed by automatic execution of a sequence of download/install/uninstall/execute steps, or it may require the consumer to access physical media such as floppy disk or CD-ROM to perform an install under direct supervision. It may involve automatic

execution, or execution under user control, following instructions indicated for the user by the advisory.

- consumer invokes Script file for solution. The advisory may offer a series of instructions in a high-level system-affecting language, such as AppleScript, DOS Shell, UNIX Shell, Visual Basic, which the consumer is expected to store as a file and then pass to a standard interpreter (*e.g.* AppleScript Editor, DOS Command Line Interpreter, UNIX Shell Command Line Interpreter, or Visual basic Interpreter). This action may alternatively involve the consumer executing a series of manual operations on the computer that involve typing in commands one by one in a certain window of a certain application.

Many concrete outcomes can be grouped among the outcomes in this list.

Advisory File Format

The advisory file format provides a mechanism to encode one single advisory or several advisories for transport across computer networks and other digital transport media, and to offer one or several variants of same basic explanatory material. The following discussion describes the components of an advisory in general terms and describes the currently understood best method for implementing advisories using MIME.

Components of a Basic advisory

The most elementary advisory may have these logical components (see Fig. 8):

- Wrapper. Components designed to package the information for transport and subsequent decoding.

- From Line. Component identifying the advice author.

- Subject Line. Component identifying briefly the concern of the advisory.

- Relevance Clause. Component in the formal relevance language precisely specifying the conditions under which the advisory could be relevant.

- Message Body. Component providing explanatory material potentially explaining to the user what condition has been found relevant, why the user is concerned, and what action is recommended.

- Action Button. Component providing the user the ability to invoke an automatic execution of the recommended action.

Clause Variations

Elaboration on the basic scheme may also be valuable:

- The advisory may contain an expires-when clause. This is an expression in the formal relevance language which causes the message to expire if it evaluates to True.

- The advisory may contain an evaluate-when clause. This is an expression in the formal relevance language which causes the message to be evaluated for relevance if it evaluates to True.

- The advisory may contain an requires-inspector-library clause. This may give the name of an inspector library and a URL where it can be found. This

indicates that a certain inspector library must be installed for relevance to be evaluated correctly.

- The advisory may contain a refers-to clause, giving keyword labeling of systems referenced by the condition associated with the advisory.

- The advisory may contain a solution-affects clause, giving keyword labeling of possible effects of the recommended response.

Other variations may be recognized as useful in the future. Such variations are not excluded from the scope of the invention.

Display Variations

The message body may occur in at least three forms:

Text. The explanatory material may be an unconstrained ASCII text document. This has no embedded variations in presentation style (*e.g.* no changes in font and/or no hypertext references to outside documents).

HTML. The explanatory material making up the message body may be an HTML document. This is familiar from Web browsers. HTML documents may contain variations in the presentation of text, may contain tables and visual formatting features, may contain references to external documents, and may contain references to external graphics files.

Text/HTML. The explanatory material making up the message body may be given in both text and HTML forms. The advice reader has the option of using whichever form is more appropriate to the user.

Further variations in message content, including audio and video content, are not excluded from the scope of the invention.

Digital Integrity and Authenticity

The message body may have digital authentication features appended to the message to insure its integrity and authenticity.

A digital digest may be appended to the message to ensure message integrity. At the time that the message is compiled by the author, a specialized functional of the message body may be computed and appended to the message. The recipient of the message can verify the integrity of the message by computing the same functional and verifying that it produces the same result as that appended to the message. Known examples of digital digests include CRC, MD5, and SHA.

Digital digests are familiar in the computer programming community under the name hashing. The idea is that certain mathematical operations based on modular arithmetic are applied to a numerical representation of a body of text, producing a numerical output ranging in magnitude from a small number to a number requiring some dozens of digits to represent, depending upon the details of the digest mechanism. These arithmetic operations typically produce an output which depends on the original body of text in a discontinuous way which is not easily invertible. That is, slightly different messages tend to have very different digests. Also, it may be difficult to find any two messages with the same digest, and if one of the two messages is previously specified, it is particularly difficult to find another message which happens to have the same digest.

The practical implication is that a transmission or recording error which causes the advisory document to be modified in some way from the authors original intent does not typically result in a modified document that generates the

appropriate digital digest. In this way, modified documents can be identified and suppressed from consideration.

A digital signature may be appended to the message to ensure message authenticity (see C. Pfleeger, Security in Computing, Second Edition, Prentice-Hall (1996); and PGP 4.0 Users Manual, PGP Pretty Good Privacy, Inc. (1997)). This is a refinement of the digital digest idea, rendering the digest secure against malicious tampering.

Digital signatures generally work as follows: At the time that the message is compiled by the author, a digital digest of the message is calculated. The digest is then encrypted using an encryption scheme that is well known and widely associated with the advice site. The encrypted digest is considered the advice site's signature on the message, and is appended to the message itself, labeled as a signature.

The advice reader, in seeking to verify the signature of the site, attempts to decrypt the signature using the well-known decryption algorithm associated with the advice site. A successful decryption produces a digital digest which agrees with the value that the advice reader calculates directly from the message. An unsuccessful decryption produces a result that does not agree with the digital digest of the received message.

It is commonly accepted (see C. Pfleeger, Security in Computing, Second Edition, Prentice-Hall (1996); and PGP 4.0 Users Manual, PGP Pretty Good Privacy, Inc. (1997)) that this approach, when used in conjunction with certain well-know encryption systems, produces a secure digital document. That is, it is accepted that a malicious agent cannot easily modify a given valid advisory to produce an impostor advisory which produce a successful decryption.

Indeed, to deceive this system successfully, it is necessary for the impostor to generate the digital digest of the modified document correctly and then apply the encryption algorithm associated with the advice site. While the impostor may be assumed to have learned the workings of the digital digest mechanism, it is assumed that he is not able to encrypt documents as if he were the advice site.

The fundamental assumption of modern cryptography systems as applied to public communication is that certain encryption/decryption algorithms can have widely known decryption algorithms and keep the encryption algorithms secret. Until this fundamental assumption is disproved, the digital signature mechanism is widely considered an effective authentication mechanism.

MIME

In the currently understood best method for structuring advice for Internet transport, an advisory document is packaged as a single ASCII text file which is a valid instance of MIME file (see N. Borenstein, N. Freed, MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, Internet Standards Track RFC 1521 (1993)). Actually, only a special subset of the full MIME format is used. Special extensions to MIME are added to accommodate the invention.

MIME is an Internet standards track format extending the classical e-mail Internet standard commonly referred to as RFC 822. The MIME format is widely used for Internet transport of electronic mail. It has four features of particular usefulness in connection with advisories:

Header Lines. MIME specifies that a message body may be preceded by an extensive message header consisting of a variety of header lines, where individual lines begin with a well known phrase and contain addressing, dating,

and related commentary. Some of these lines can be easily adapted to serve the purposes of the invention. For example, the From Line and Subject Line components of an advisory can be implemented by the From: and Subject: header lines that are already part of the MIME standard.

Extensibility. MIME provides a method for creating new message lines in messages. This includes a method for embedding the new message lines in messages and a method for registering the new line with the MIME authorities. Key invention constructs relevant-when and expires-when may therefore easily be added to the MIME language in that fashion.

Alternation. MIME provides a method, *i.e.* Multipart-Alternate, for offering two different versions of the same message, with the destination picking the appropriate display method. Therefore, the invention construct of transmitting one or more ways to display the same information may easily be implemented using the MIME standard and its Multipart-Alternate feature.

Digesting Mechanism. MIME provides a well-understood mechanism, *i.e.* Multipart/mixed, for packing several complete MIME messages into a single file for Internet transport. MIME posits a recursive digest structure, in which a message can have several related components, and each component can itself be a MIME file inserted verbatim. Using this feature, a MIME file can be used to digest many component advisories, organized in a tree structure reminiscent of the branching structure of a modern personal computer file system.

Thus, MIME becomes a tool, not for packaging e-mail, but instead for packaging a new kind of document, *i.e.* the advisory. To avoid confusion, it should be appreciated that an advisory is unlike e-mail because an advisory does not have an intended recipient or list of recipients. Rather, it is a broadcast message. An advisory typically has relevance and related clauses, and an advisory typically

has active content. E-mail does not have relevance and related clauses, and does not typically have active content. The advisory is part of a new form of communications which can be implemented within the MIME standard. The advisory application of MIME addresses a different problem than e-mail by omitting certain MIME clauses which were used for e-mail, and by adding new specialized clauses which are used in the relevance determination and advice management process. In a certain sense, the relationship of advisories to e-mail is comparable to the relationship between USENET and e-mail. Both advisories and USENET news systems use MIME as a packaging mechanism. However, both offer means of communications which are distinct from e-mail.

Although MIME is a convenient method of realizing the form of an advisory, there is no necessary connection of the invention to MIME. There are many other common formats in the Internet world, such as XML, which may be used for representing advisories. In this disclosure, only the currently understood best method for implementing the advice file is discussed.

Example

The following is an example of an advisory file:

```
Date: Sat Mar 21 1998 17:06:12 +0800
From: Jeremiah Adviser <jeremiah@advisories.com>
MIME-Version 1.0
Organization: Universe Communications, Inc.
Subject: A better version of the advice reader is now available
relevant-When: version of application "advice.exe" < version "5.0"
Content-Type: text/html; charset=us-ascii

<HTML><BODY>
```

A better version of the advice reader is available.
Click to <A HREF="http://www.advisories.com/win98/advice50.exe">
Download </A> the latest version of advice reader.
</BODY></HTML>

Here the reader can see the various components of an advisory embodied as MIME components:

- Wrapper. MIME-Version and Content-Type header lines.

- From Line. From: Jeremiah Adviser ...

- Subject Line. Subject: A better version of ...

- Message Body. An HTML fragment, beginning <HTML> and ending </HTML>.

- Action Button. Not present in this advisory. The active component of the message (downloading) is handled by the HTML HREF link. The user sees the word Download and typically understands that a mouse click on that word causes the indicated action.

Ratings Blocks

In an additional variation, it is possible for an advisory to contain ratings blocks containing information rating the advisory according to criteria such as privacy, security, and usefulness. There exist standard formats for such ratings blocks (see Khare, Rohit, *Digital Signature Label Architecture*, The World Wide Web Journal, Vol. 2, Number 3, pp. 49-64 Oreilly (Summer 1997),

http://www.w3.org/DSIG) and these are easily appended to messages with MIME structure. See also below.

Relevance Language

Advisories have a format resembling the format of e-mail messages, with many of the same components in the message/digest headers. One key extension offered by advisories is the institution of a new clause in the message, *i.e.* the relevance clause. The relevance clause is preceded by the keyword phrase relevant-When:. An expression from the relevance language follows the keyword. The following discussion describes the currently understood best method for describing the state of a consumer computer.

Descriptive Language

The purpose of a relevance clause is to examine the state of an individual computer and determine whether it meets various conditions which combine to imply the relevance of a certain advisory.

In the currently understood best method for implementing the invention, the language itself, *i.e.* in the allowable phrases of the language and the underlying semantics of the phrases, provides an intellectual model of the components of the consumer computer, its peripherals, storage devices, files, and related concepts. This is distinct from the usual model of computer languages, in which the language itself provides a rather meager picture of the problem it is used to address.

In common with traditional languages, the relevance language contains a few elementary data types, such as Boolean, integer, and string. Also in common

with traditional languages, it is permissible to write arithmetico-logical expressions such as:

(2346 + (-1234) / (1 + 2)) > 0

The meaning of a typical subexpression, *e.g.* 1+2, is apply method + to the pair of objects resulting from evaluating the two subexpressions 1 and 2. The pair of objects in question are objects of type Integer having values of 1 and 2, respectively. In the currently understood best method, the relevance language has a full range of arithmetic, string, and logical operations available, which are expressed as built in methods set to operate on the built in concrete data types (see Fig. 12).

Unlike traditional languages, the relevance language contains an abstract data type, World, which may be thought of as the overall environment of the personal computer on which the relevance clause is evaluated. This object has properties. These properties yield objects of various types, and these objects may have further properties (see Fig. 13).

World is a data type that, depending on the specific implementation and on the specific system configuration, may have many properties.

In the technical support application discussed above, these properties may include the system folder property, the CPU property, and the monitor property. Properties of an object are obtained by applying assessor methods to the object. The assessor method for the system folder of data type World returns an object of type system folder. The assessor method for the CPU property of data type World returns an object of type CPU. These derived objects, in turn, have properties of their own. For example, an object of type CPU may have a collection of properties such as speed, manufacturer, model, MMX, and cache. A

method corresponds to each of these properties which, when applied to the object of type CPU, returns a result. For sake of discussion, it can be assumed that speed results in an integer, manufacturer results in a string naming the manufacturer, model results also in a string, naming the model type, and MMX and cache return the more specialized object types MMX, and cache.

The relevance language implicitly postulates that the set of inspectable properties of the consumer computer is identical to the set of properties of data type World and the set of properties derivable from World by repeated applications of asking for properties of an object derived from World (see Fig. 14). ObjectWorld gives an idea of the richness of the object world derivable in this way in the technical support application.

Example Relevance Clauses

The following are examples of relevance clauses as used in a technical support application:

Existence of a certain application on the consumer computer

relevant-When: exists application "Photoshop"

The intent of this fragment is that application is a property of World which takes an extra string parameter and returns an object of type application. exists is a property of any object, which returns the Boolean True if the object exists. If the application named Photoshop cannot be found by the method implementing the application property, then the result is a non-existent object, for which exists returns the Boolean False.

Comparison of version numbers

relevant-When: version of Control Panel "MacTCP" is version "2.02"

The intent of this fragment is that Control Panel is a property of the World which takes an extra string parameter and returns an object of type Control Panel. If the Control Panel named MacTCP cannot be found by the method implementing the Control Panel property, then the result is a non-existent object, for which version is not an allowed property, and evaluation fails. If the Control Panel named MacTCP is found, then version, being an allowable property of Control Panels, leads to invocation of a method which returns an object of type version containing the version number of that Control Panel, recorded in a particular format. This result is compared with the result of subexpression version "2.02". This time version refers to a property of World, which takes an extra string parameter and returns an object of type version. If evaluation succeeds, the result of this comparison is Boolean: either True or False.

Compare modification dates

relevant-When: modification time of Photoshop PlugIn "Picture Enhancer" is greater than time "10 January 1997 12:34:56 +0800"

The intent of this fragment is that Photoshop PlugIn is a property of the World which takes an extra string parameter and returns an object of type Photoshop PlugIn. If the Photoshop PlugIn named PictureEnhancer cannot be found by the method implementing the Photoshop PlugIn property, then the result is a non-existent object, for which modification time is not an allowed property, and evaluation fails. If the Photoshop PlugIn named PictureEnhancer is found, then modification time, being an allowable property of a Photoshop PlugIn, leads to invocation of a method which returns an object of type time. This result is

compared with the result of subexpression time "10 January 1997". Here, time refers to a property of World which takes an extra string parameter and returns an object of type time. If evaluation succeeds, the result of this whole expression is Boolean: either True or False.

Automatic Parsing and Evaluation

A key purpose of the relevance language is to enable an advice provider to publish advisories which can be accessed by the advice reader, running on a consumer computer, and be automatically read to determine, without intervention from the consumer, whether the advisory is relevant to the consumer.

In the currently understood best-method, the relevance language is implemented as a context free grammar which can be automatically parsed into a tree of subexpressions. The tree of subexpressions can be understood as an abstract structure whose nodes are methods and whose branches are subexpressions.

This tree is represented using a standard notation in computer science:

(node (expr-1) (expr-2) .... (expr-n))

where node gives the name of the method to be applied, and (expr-k) stands for the k-th subexpression to be furnished to the method. For example, the expression:

(2346 + (-1234)/(1 + 2)) > 0

can be parsed into the expression tree:

```
(>      (+      (Integer 2346)
                (/      (Integer -1234)
                        (+      (Integer 1) (Integer 2))
                )
        )
        (Integer 0)
)
```

The expression:

exists application "Photoshop"

can be parsed into:

(exists (application "Photoshop"))

The expression version of Control Panel "MacTCP" is version "2.02" parses into:

```
(is     (version  (Control-Panel  "MacTCP"))
        (version  (string "2.02"))
)
```

Finally, the expression:

modification time of Photoshop PlugIn "Picture Enhancer" is greater than time "10 January 1997"

parses into

```
(is-greater-than    (modification-time (Photoshop-PlugIn "Picture Enhancer"))
                    (time (string "10 January 1997"))
)
```

In short, the goal of parsing is to identify a sequence of method invocations to be applied. Procedures for parsing context-free grammars into expression trees are well-understood (see A. Aho, J. Ullman, Principles of Compiler Design, Addison-Wesley (1977)). A lexer breaks the input into a series of tokens. In the currently understood best method, these tokens may take of the following forms:

[String] A string of printable ASCII characters enclosed in quotation marks (").

[Integer] A string of decimal digits.

[Minus] The character -.

[SumOp] The characters +-.

[PrdOp] The characters */ and the string mod.

[RelOp] The character sequences = > >= <= != and the relational phrases and or is not.

[Phrase] A sequence of one or more unquoted words, a word being an alphanumeric string beginning alphabetically and not containing embedded blanks. Phrases break at reserved phrases.

Parsing proceeds mechanically according to a precedence table giving the productions of a grammar. In the currently understood best method, the productions in the grammar are as follows:

```
<Goal>          := <Expr>
<Expr>          := <Expr> or <AndClause> | <AndClause>
<AndClause>     := <AndClause> and <Relation> | <Relation>
<Relation>      := <SumClause>[RelOp]<SumClause> | <SumClause>
<SumClause>     := <SumClause> [SumOp] <Product>
                | <SumClause> [Minus] <Product>
                | <Product>
<Product>       := <Product>  [PrdOp] <Unary>
                | <Unary>
<Unary>         := [Minus] <Unary>
                | [UnyOp] <Unary>
                | <Cast>
<Cast>          := <Cast> as [Phrase]
                | <Reference>
<Reference>     := [Phrase] of <Reference>
                | [Phrase] [string] <Restrict> of <Reference>
                | [Phrase] [integer] <Restrict> of <Reference>
                | [Phrase] [string]  of <Reference>
                | [Phrase] [integer]  of <Reference>
                | [Phrase] <Restrict> of <Reference>
                | [Phrase] [string]
                | [Phrase] [integer]
                | [Phrase] <Restrict>
                | [Phrase]
                | exists <Reference>
                | number of  <Reference>
```

| [string]
| [integer]
| it
| (<Expr>)

<Restrict> := whose (<Expr>)

In this display, word stands for a reserved word in the language, [Phrase] stands for a phrase as defined in the discussion of lexical analysis on the previous page.

A grammar can be used to generate a parser by any of several means (see A. Aho, J. Ullman, Principles of Compiler Design, Addison-Wesley (1977)). These may include automatic parser generators, such as YACC, which create a table driven finite state automaton that recognizes the grammar. The table is created directly from the production forms above, and also by hand generation of recursive descent parsers based on mimicking the productions of the grammar in modules whose naming and internal structure mimic the structure of the productions of the grammar.

All such approaches have the same basic result. New tokens are input, one-at-a-time, and compared with the current state and also with a table giving allowable type and mandated action on receiving that token, if any. The mandated action can be interpreted as specifying the individual steps in the systematic building up of an expression tree. A typical action is that associated with the production:

<Relation> := <SumClause> [RelOp] <SumClause>

which could be written, in a standard notation, as:

$$ = ($2 $1 $2)

This is interpreted as follows: $$ refers to the result of the production, $1, $2, $3 refer to the component subexpression trees, and the parentheses are notational devices that are used to delimit expression trees. This action calls for the association of the recognized <Relation> with an expression tree. This results from joining expression trees which are associated with the left-subexpression and the right sub-expression with a root method that compares the two expressions. Consider the expression version of Control Panel "MacTCP" is version "2.02". Consider the state of the parser at the moment that it attempts to apply the <Relation> production with [RelOp]. The expression tree already associated with the left subexpression, $1, has representation (Control-Panel "MacTCP") and that associated with the right subexpression, $3, has representation (version (string "2.02")). The expression tree associated to the overall <Relation> expression is the merger of these two according to the pattern (is $1 $3). Hence, the resulting expression tree is representable as (is (Control-Panel "MacTCP") (version (string "2.02"))).

Associated with each production is an action of appropriate form which describes how the tree is built. In certain implementations, the tree may only be built up implicitly.

Parsing can continue normally, if at every step of the parsing the next available symbol matches an allowable type; or it can fail, if an unexpected combination occurs. As soon as parsing fails, the piece of advice may be declared not relevant.

In the currently understood best method of implementing the invention, each valid method is already known to the parser at parse time. Unlike some other languages, parsing can fail if a clause is syntactically correct but uses phrases that name currently unknown methods.

In the currently understood best method of implementing the invention, each subexpression takes values which are strongly typed and for which the type is known in advance. Example data types include integer, string, and Boolean. Each method is known at parse time to work with certain combinations of data types of inputs and to give certain definite data types as outputs. Attempts to apply methods to forbidden data types are diagnosed as failure of the parse. If so, the piece of advice may be declared not relevant.

At the successful completion of parsing, an expression tree is built up consisting in essence of a collection of method invocations and associated arguments and associated data types of those arguments. Evaluation of the expression is the process of performing the appropriate method dispatching in the appropriate order.

Evaluation can be successful, or it can fail. It can fail, for example, from excessive use of system resources, unavailability of a resource, excessive delay in obtaining a resource, or for some other reason. Successful evaluation can yield a Boolean value of True or False or some other value. The interpretation of a piece of advice as relevant is equivalent to saying that the evaluation is successful, the value was Boolean, and is true.

In particular, if a certain subexpression cannot be interpreted as a valid expression in the language, if the subexpression attempts to apply methods to forbidden data types, or if the subexpression cannot currently be evaluated, the whole expression can fail, and the advice is automatically declared not relevant.

Extensible Language

The purpose of the relevance language is to describe precisely the state of a computer, its contents, attachments, and environment. This state can change as the consumer purchases new software and/or hardware, or as new software/hardware objects are invented. This state can change as consumer computers are used to represent consumers in new problem areas, for example, in personal finance, management of communicating devices in the home, or other areas.

Consequently it is not possible to delimit in advance the components of state that may be of interest to which the invention provides access. It is desirable for the relevance language to give future authors the ability to extend the relevance language to express concepts about system state that have not yet been conceived.

In one implementation of the invention, the vocabulary of the relevance language may be extended by the authorities and by authors at individual advice sites.

In that implementation, the relevance language is extensible by developing dynamically loaded libraries which add new vocabulary and semantics to the language and/or modify existing vocabulary and methods. These are referred to herein as inspector libraries and may be downloaded from an advice site and installed on a given consumer computer, thereby changing the meaning of the relevance language on that computer, and allowing new bodies of advice to be interpreted on that computer.

These dynamically loaded libraries contain declarations of the new data types which must be added to the language, of the new properties associated with the data types, of the data type resulting when a specific property is obtained for an

object of a specific type, and of methods, *i.e.* executable code that implements access to the properties.

Non-Procedural Language

Unlike many languages used in connection with the operation and/or maintenance of computers, the relevance language does not need to be procedural. That is, it need not specify how to manipulate the contents of various fragments of memory. This is the opposite of being descriptive. It is not necessary to enable traditional procedural services, such as loops, assignments, and conditionals.

On the contrary, making these services available in an expansive fashion may pose various security and privacy threats, by making it easy for carelessly written or maliciously written advisories to consume excessive resources at evaluation time.

In the currently understood best method of implementing the invention, procedural services are not made available in the relevance language. As inspection of the above grammar description shows the language has:

- no named variables
- no assignment statements
- no function calls, or at least no explicit function calls with variable arguments
- no loops or conditional execution

These differences in appearance between the relevance language and other common languages are rooted in the following view:

- Because of concerns about unattended evaluation, the language should ideally have no side effects on the computer or environment.

- To inspire consumer confidence, consumers must have be able to see for themselves that the language has no effects on the computer or environment.

- A descriptive language, unlike a procedural one, has the appearance of having no side effects.

In short, the structure of the language and the visible limitations should communicate a message of security to the consumer.

The following discussion addresses two key differences of the relevance language from procedural languages:

Function Calls. The relevance language has method dispatches which correspond to function calls in some other languages, but they are of a more tightly constrained form.

First, there are the unary methods and the binary methods that occur in arithmetic and logical operations: +,-,*,/, and, or, =, and similar operations. These can be thought of as unary or binary function calls, but they are of a very restricted form, implementing well understood methods that typically pose little danger or resource burden.

Second, there are unnamed properties such as modification time.

Third, there are named properties such as application Photoshop".

The unnamed properties can be thought of as function calls applied to an object, but very bland ones, because no parameters are involved. Typically, a property is computed by extracting a certain value from a certain slot of a data structure. They typically pose little danger or resource burden. The named properties may be thought of as two-variable function calls. The first variable is the object and the second object is the string name-specifier. However, these also are not very general operations because the string name-specifier, in one implementation, may not itself be a computed result. It must instead be string constant. The types of calculations that can be specified in this way are tightly constrained. Again, typically a named property is computed by extracting a certain value from a certain slot of a data structure, so it poses little danger or resource burden.

Loops and Conditional Execution.

The relevance language has no for, while, or if statements, but it does have a limited ability to perform iteration. It does this using a construct referred to as plural properties. In the relevance language there can be both singular and plural properties, *e.g.* both entry and entries properties, the first referring to a result which must be a singleton and the second referring to a result which may be a plurality. Typically, pluralities are further qualified by the use of the whose () clause to restrict to subcollections.

By the plural-singular dichotomy, certain fine distinctions of meaning may be maintained. For example:

exists application "Photoshop"

has the meaning that there exists exactly one such application; and

exists applications "Photoshop" whose(version of it is version "4.0")

has the meaning that there exists one or more than one application called "Photoshop", and among those there exists one with version 4.0.

In the second example, an iteration is implicitly performed over the collection of all applications called Photoshop" on the system in question, so the effect of a loop is obtained without using traditional procedural programming.

The restrictions on the expressiveness of the language help make the language safer from the viewpoint of privacy and security guarantees (see below). Nevertheless, the language is designed to be powerful in that it is intended to be highly expressive. A few words in this language provide access to answers about the system state which would be impossible to obtain in traditional procedural languages short of writing hundreds of lines of code and invoking many specialized functions in system libraries.

If an apparent need should arise for the kind of services that traditionally are handled by procedural languages, it may typically be satisfied by extending the relevance language using the inspector library mechanism mentioned earlier, and described in more detail below. This has two advantages:

[Efficiency] Including new inspectors by this extension mechanism, rather than by offering procedural services in the relevance language, leads to more efficient execution. Inspectors typically make available efficient compiled methods of execution, minimizing burden on system resources at relevance evaluation time while the relevance language is interpreted, which is typically slower.

[Security] Including new inspectors by this extension mechanism allows one to correct problematic situations. If a certain complex expression is used in many places and has bad side effects, then it can be very hard to correct. If an equivalent piece of code is included as an inspector library, then one can identify the problematic code by using the relevance language to identify whether that inspector is installed on the user computer. This makes it possible to write counter advisories against advice that depends on inspector libraries.

Consumer-Accessibility

The relevance language controls the execution of a system on a potentially vast number of computers. It is highly desirable, though not strictly necessary, for a relevance clause to be something which, in principle, a consumer could read and form an approximate understanding of, though few users may choose to do this in most cases.

In the currently understood best method of implementing the invention, the syntax of the relevance language resembles the syntax of plain English, with key roles in the language played by clauses formed from articles such as of, as, whose, and verbs such as exists.

The highly constrained nature of the language fosters consumer understanding. The language avoids constructs which assume a computer programming background by suppressing concepts such as arrays, loops, and conditional evaluation.

Inspector libraries

Components of inspector libraries

Parsing of a clause in the relevance language results, conceptually, in the generation of a list of method dispatches (see Fig. 11), in which certain methods are called in a certain order with certain argument lists. This evaluation is a process of systematically carrying out the sequence of method dispatches in the appropriate order. Method dispatches are an important aspect the relevance process.

An inspector library is a collection of methods (see Fig. 15) and associated interfaces which allows for the installation of methods into the advice reader. Because of the structure of the parser and the evaluation process, an inspector library may contain some of the following components:

- Declaration of a [Phrase] to be used in the relevance language.
- Association of that [Phrase] to a specific method.
- Declaration of a new data type to be used in the evaluation process.
- Declaration of the calling prototype of the method. This includes the number and the required data types of the arguments to be supplied to the method.
- Declaration of the result data type of the method.
- Implementation of that method in executable form.

- Declaration of special hooks associating code to be called on events, such as advice reader initialization, advice reader termination, beginning of advice reader main evaluation loop, and ending of advice reader main evaluation loop.

- Declaration of special hooks associated with creation and maintenance of special caches associated with the method.

- Implementation of special event methods and cache methods in executable form.

Conceptually, an inspector library can be linked into the advice reader with all the declarations evaluated, resulting in changes to the advice reader's internal data structures, so that new method invocations become available.

These declarations affect two fundamental data structures of the system. The first is a syntax table giving all allowed phrases and the associated data types on which they may operate and the associated data types that result. This is used at lexical analysis time. The second is a method dispatch table, giving a systematic way to determine the associated executable method for a given phrase and data types. This is used at evaluation time.

Object-Oriented Structure

A convenient way to implement the above inspector library structure is to rely on the features of a modern object-oriented programming language, such as C++. In effect, the built-in features of such a language, *i.e.* object declarations, polymorphism, and operator overloading, are ways of declaring that certain phrases have a certain meaning when applied to certain data types, and of systematically organizing that information. Other features, such as constructors,

copy-constructors, and destructors, are ways of defining certain initialize time and terminate time code bodies.

In the currently understood best method, such features of modern object oriented languages are used to provide the various features of inspector libraries.

Extension

In one implementation, as described above, it is possible to install several inspector libraries in an instance of the advice reader. The inspector libraries that are so installed define the set of recognized [Phrase]s in the relevance language, the set of allowable data types at evaluation time, and the set of methods associated with those data types.

In short, the relevance language may be dynamically constituted. In one implementation, inspector libraries may be created by advice providers and downloaded to the consumer computer as part of the site synchronization. Such libraries may be managed by the advice reader, for example, by storage in a well known location, such as a subdirectory of the overall directory managed by the advice reader. The inspector libraries in this directory may be linked into the advice reader at the time the advice reader is initialized.

When this linking happens, declaration routines are invoked, installing new [Phrase]s in the lexical analysis table of the relevance language, and associating these [Phrase]s to certain method invocations. The language expands in this way to include new descriptive possibilities.

Layered Language Definition

The relevance language may therefore be open ended, built up in layer upon layer of extensions. Hence, to understand a completely installed system is to understand the layers which have been installed, and to understand the methods that each layer provides. In a typical installation, these layers are as follows:

- Base Layer. Contains the basic mechanics of clause evaluation: a number of basic built-in phrases and associated methods. It is expected that the base layer is the same on every consumer computer carrying the advice reader.

- System-Specific Layer. This consists of a layer associated with a certain operating system, giving information about the characteristics of a certain family of computers and their attached devices and environment. For example, such a layer, in one implementation, provides methods to get the system date and time, the sizes of various files, the contents of the PRAM, or the names of attached peripheral devices.

- Vendor-Specific Layers. This collection of potentially a large number of extensions layers is typically produced by third parties, giving special access to the internals of certain hardware devices and software products. One can think of potential authors ranging a span of products from hardware producers (*e.g.* of cable modems) to software producers (*e.g.* of Photoshop and plug-Ins) to service providers (*e.g.* America On-Line).

Example: Version inspector

The following is an example of an inspector for the version property of data type Application under the Macintosh OS. This inspector declares the following:

- A new [Phrase] to be added to the relevance language: version;

- A new data type, version, which has already been referred to in several examples above;

- Several properties of this data type which are available under Macintosh OS:

  Major Revision. The leading numeric field of the revision number.

  Minor Revision. The secondary numeric field of the revision number.

  Stage. A String, such as Alpha, Beta.

  Country. A String, such as USA or France.

  String1. A String.

  String2. A String.

- Methods, in the form of executable code, which implement the above properties by opening the resource fork of the application, extracting the desired information, and converting into the required data types.

- A new named property of World, version, which casts a string property specifier, such as the 1.1 in version 1.1, into a version data type.

Upon installation, this inspector makes available to the system a series of data types and properties which may be as depicted in Fig. 14. As an example, to

check if the beta version of an application with version number 0.99 is used, one might write the relevance clause:

Stage of application "Netscape Navigator" is "Beta"
and Minor Revision of application "Netscape Navigator" is 99
and Major Revision of application "Netscape Navigator" is 0

Special Inspectors

The language extension mechanism described above has powerful consequences, for example, as described in the following:

OS Inspectors

A system specific inspector can access the properties of the operating system and allow advice to be written to verify the existence and configuration of attached devices and other subsystems.

The following is an example of a valid fragment written for use with the Macintosh OS inspector library:

exists serial device "Modem Port"

The intent of this fragment is to check if this is the type of Macintosh having a dedicated modem port, which is to be distinguished from a Modem/Printer Port. The property of World referred to as serial device potentially matches several different devices. The qualifier selects from among those the one which has the name "Modem Port." If there are any such devices, the phrase evaluates to True. If not, the phrase evaluates to False.